# So many factors, so little time...

# Simulation experiments in the frequency domain

Susan M. Sanchez[a,*,1], Farhad Moeeni[b] and Paul J. Sanchez[c]

[a]*Operations Research Department and Graduate School of Business & Public Policy, Naval Postgraduate School, Monterey, CA, 93943, U.S.A.*

[b]*Department.of Computer & Information Technology, Arkansas State University, P.O. Box 130, State University, AR, 72467, U.S.A.*

[c]*Operations Research Department, Naval Postgraduate School, Monterey, CA, 93943, U.S.A.*

We illustrate how an efficient methodology called frequency domain experimentation can be used to gain better insight into the behavior of production systems. With the full factorial designs commonly used for simulation experiments, the number of runs grows exponentially with the number of factors involved, while the run length remains constant. In frequency domain experiments, the number of runs is independent of the number of factors, while the required run lengths increase relatively slowly. We describe the method, illustrate its effectiveness at identifying important main effects, two-way interactions, and quadratic terms for a known model, demonstrate the approach by evaluating a kanban system involving 34 factors, and provide links to software. We also present computational requirements for running simulation experiments that combine a batch means approach with efficient run-oriented designs for a variety of systems. The results indicate that frequency domain experiments perform very favorably for systems, such as queueing networks, where the simulation's output stream exhibits high positive autocorrelation.

*Keywords:* Just-in-time manufacturing, Production planning, Design of experiments

[*] Corresponding author: Tel: +1-831-656-2780; fax: +1-831-656-2595

*E-mail addresses:* ssanchez@nps.edu (S. M. Sanchez), moeeni@astate.edu (F. Moeeni), PaulSanchez@nps.edu (P. J. Sanchez)

Received December 16, 2004; revised April 7, 2005; accepted June 16, 2005.

# Report Documentation Page

*Form Approved*
*OMB No. 0704-0188*



| | | |
|---|---|---|
| 1. REPORT DATE<br>**7 APR 2005** | 2. REPORT TYPE<br>**N/A** | 3. DATES COVERED<br>**-** |

| | |
|---|---|
| 4. TITLE AND SUBTITLE<br>**So many factors, so little time... Simulation experiments in the frequency domain** | 5a. CONTRACT NUMBER<br>5b. GRANT NUMBER<br>5c. PROGRAM ELEMENT NUMBER |
| 6. AUTHOR(S) | 5d. PROJECT NUMBER<br>5e. TASK NUMBER<br>5f. WORK UNIT NUMBER |
| 7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)<br>**Naval Postgraduate School Operations Research Department Monterey, CA 93943** | 8. PERFORMING ORGANIZATION REPORT NUMBER |
| 9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) | 10. SPONSOR/MONITOR'S ACRONYM(S)<br>11. SPONSOR/MONITOR'S REPORT NUMBER(S) |

12. DISTRIBUTION/AVAILABILITY STATEMENT
**Approved for public release, distribution unlimited**

13. SUPPLEMENTARY NOTES

14. ABSTRACT

15. SUBJECT TERMS

| 16. SECURITY CLASSIFICATION OF: | | | 17. LIMITATION OF ABSTRACT | 18. NUMBER OF PAGES | 19a. NAME OF RESPONSIBLE PERSON |
|---|---|---|---|---|---|
| a. REPORT<br>**unclassified** | b. ABSTRACT<br>**unclassified** | c. THIS PAGE<br>**unclassified** | **SAR** | **35** | |

**Standard Form 298 (Rev. 8-98)**
Prescribed by ANSI Std Z39-18

## 1. INTRODUCTION

Simulation is a useful tool that one can apply in diverse areas, including the design and control of manufacturing facilities, the evaluation of hardware or software requirements for computer networks, the analysis of financial or economic systems, and the design and analysis of transportation systems. Since it is often less costly and time consuming to experiment with a simulated system than to do so with a real-world system, one can use simulation prospectively to design or analyze the performance of systems that do not currently exist.

Consider the many phases of a full-scale simulation project, as illustrated in Figure 1. As one abstracts the real-world or prospective system and then implements it as a functional simulation program, one must conduct several distinct types of activities. The programming and debugging tasks in the verification stage can be time-consuming and difficult, particularly for highly complex models. Validation ensures that the simulation output adequately approximates the performance behavior of the true system. Validation efforts that establish the credibility of the conceptual model as a representation of the real-world or prospective system are necessary if the results of the simulation are to gain respect and play a role in managerial decision-making.

*** Figure 1 about here ***

Because of these difficulties, many perceive simulation inappropriately as an exercise in computer programming rather than as model building and analysis. As a result, the exploration/experimentation phase may receive short shrift. A large number of factors and the presence of nonlinear effects as well as multi-factor interactions may affect overall performance, compounding this oversight. Despite advances in computing speed, large amounts of computer time may be required to develop an adequate representation of the system's behavior.

In this paper, we demonstrate how the methodology called frequency domain experimentation (FDE) developed by Schruben and Cogliano (1987) can provide insights into the behavior of complex production systems. An appropriately designed FDE allows analysts to simultaneously examine a large number of potential factors, quadratics, and interaction effects to determine their impact on the system's response. By providing a concise description of the procedure, along with access to computer programs for designing and analyzing FDEs, we hope to remove technical hurdles that might otherwise have prevented researchers and analysts from implementing FDEs. FDEs require fewer runs than competing techniques, and use less data when the simulation's output stream is highly correlated.

We begin with a discussion of methods for exploration and experimentation, and motivate the need for efficient experimental designs in Section 2. We describe the FDE methodology in Section 3, and illustrate its use for two different

types of systems in Section 4. The first is a simple stochastic system in which the true behavior is known and hence the FDE results can be compared to the underlying model; the second is a just-in-time (JIT) system using kanban. In Section 5, we discuss the results and compare the FDE method with other approaches for estimating main effects, two-way interactions, and quadratic terms. We recommend the use of FDEs when the analyst seeks to identify important *terms in a second-order model*, rather than solely screening for important *main effects.* Section 6 contains our concluding remarks.

## 2. EXPLORATION/EXPERIMENTATION METHODS

A well-designed experiment allows analysts to examine several prospective system configurations simultaneously. An experimental design can be viewed as a matrix with a column for each of the factors. A row in the matrix (called a *design point*) specifies one specific combination of factor level settings. Many operations management (OM) studies in the literature use full factorial experimental designs because of their simplicity, and because they allow the analyst to identify interactions among the factors as well as main effects. For example, Enns (1995) uses a 2×3×2 design to assess the impact of average utilization, load rules, and scheduling approaches on a flow shop with finite scheduling and internally set due dates. Malhotra and Ritzman (1994) consider a $2^4$ factorial design for assessing the impact of demand variability, capacity utilization, and mix and route flexibilities on postal service stations. Kim and Bobrowski (1995) examine job shop performance using a 4×4×2 full factorial for job-release mechanisms, scheduling rules, and due-date tightness levels. Vakharia et al. (1996) use a factorial design to examine six experimental factors in an investigation of the operational impact of parts commonality.

Results from a factorial experiment are often analyzed using ANOVA. Alternatively, if quantitative factors (e.g., utilization rates, lead times, capacities) determine some or all of the configurations, then the analyst can construct *response-surface metamodels* of the system performance using regression techniques. These metamodels can provide insight into the system behavior as a whole, or suggest 'good' sets of input factors for improved system performance. For example, Kleijnen (1993) uses response-surface methods to study a decision support system in a Dutch steel tube factory. While this approach is less often used in OM research than ANOVA, perhaps in part due to the qualitative nature of many input factors of interest (such as FIFO or LIFO priority rules, scheduling rules, etc.) it is a well-known approach in the applied statistics and simulation communities.

The techniques just described all have a run-oriented approach, as shown in Figure 2. Input factor levels (such as scheduling rules, machine characteristics, etc.) are constant during the course of a run. The simulation code is a 'black box' that transforms these inputs (and often pseudo-random error) into simulation output that mimics the output of the real or

prospective system. For each design point, the analyst makes one or more simulation runs and computes the average output measures.

*** Figure 2 about here ***

Run-oriented approaches work well when we vary only a few factors, or when only main effect models are of interest. For example, a full factorial experiment involving four factors requires only sixteen runs. Saturated or nearly-saturated fractional factorials require fewer runs if not all interaction effects need be estimable to construct the metamodels. A $2^{k-p}$ fractional factorial explores *k* factors (each at two levels) in only $2^{k-p}$ runs. R*esolution III fractional factorials* require the fewest runs; they are referred to as screening designs, and used to identify important factors when only main effects are of interest. Factorial and fractional factorial designs are discussed in any basic experimental design resource, such as Box et al. (1978), Montgomery (2000), or NIST/SEMATECH (2005).

Unfortunately, most problems of interest to OM researchers are not so simplistic. There may be many factors worth investigating, important interactions between factors might exist, or there might be nonlinear relationships between the factors and the response. This means that full factorials, screening designs, and two-level designs are not appropriate choices for the experiment. For example, the apparently simple JIT system we analyze later in this paper has 34 factors varied during the experiment. A full factorial experiment involving all these factors would require over 17.2 billion factor combinations—too many to incorporate in a manually controlled run-oriented approach even with the current advances in computing technology. Organizing the runs and collating the data would itself be a massive undertaking. If the ability to estimate all two-way interactions is desired, then so-called *resolution V fractional factorials* or higher-resolution designs are needed. 3-level (fractional) factorials, such as the $3^{9-5}$ used by Cabrera-Rios et al. (2002) to design a manufacturing cell for profit maximization, allow quadratic effects to be investigated as well. Central composite designs (CCDs) do this more efficiently by adding design points to (fractional) factorials. However, the statistical literature (e.g., Box et al. 1978, NIST/SEMATECH 2005) reports these only for designs involving only 11 or fewer factors, and suggest that screening designs (concerned only with main effects) be used when the number of factors is larger.

Is there a need for designs that can be used to explore interaction and quadratic effects even when the number of factors is very large? We believe the answer is an emphatic 'Yes!' For example, Jensen et al. (1996), in their study of process flexibility in a group technology environment, state that 'myriads of untested alternatives exist' and mention order review and release systems, lot-sizing methods, partitioned environments, transfer batches, and new machines as some of the many factors requiring further investigation. Krajewski et al. (1987) examine 36 factors in an investigation of the adverse

impact of environmental uncertainty on the performance of kanban systems, but in order to make the study manageable using factorial analysis they combine their factors into seven clusters. While they investigate only main effects, they suggest the need for more comprehensive experimentation in order to identify interaction effects among the subsets of the clusters. After using a factorial design to study the impact of scheduling rules and two environmental factors on outpatient clinics, Klassen and Rohleder (1996) suggest that many factors and interactions (e.g., waiting time breakdowns and client stratification, multi-server systems, circumstances that make load-sharing or expediting desirable, and type of clinic) merit further investigation.

## 3. FREQUENCY DOMAIN EXPERIMENTATION

An alternative to the run-oriented approach is the frequency domain approach, illustrated in Figure 3. If we view the inputs and outputs of the simulation runs as *time series* rather than constant (or average) values, we can then oscillate the input factors within the course of a simulation run. The idea is a straightforward one taken from classical systems theory; namely that if the input factor affects the system performance, then the output time series will oscillate at a related frequency. Alternatively, if the input factor does not affect performance, then the 'black box' will not transmit the oscillation through to the output time series.

*** Figure 3 about here ***

In practice, it is not so easy to determine oscillation relationships by eye, particularly when the simulation time series involves randomness or the number of factors is large. Additionally, the system could either dampen or magnify the magnitude of the oscillation (a phenomenon called *system gain*), and time lags could occur between the input factor variation and its appearance in the output time series (a phenomenon called *phase shift*). Consequently, we take the *Fourier spectrum* of the output time series. This partitions the overall variability in the output series according to its sinusoidal components. The spectrum of pure, uncorrelated error is flat, but complex systems often have natural cyclic behavior. Customer demand, for example, may follow daily, weekly, or annual patterns. Die wear and replacement forms another type of cyclic pattern. The nature of such cyclic behavior, whether deterministic or stochastic, will influence the shape of the spectrum. For example, the spectrum of positively autocorrelated error has large magnitudes for low frequencies, while that of negatively autocorrelated error has large magnitudes for high frequencies.

A frequency domain experiment involves two different types of runs. In the first type, called a noise run, we fix the levels of the input factors at given nominal values during the course of the simulation run, and then observe the output stream. Random fluctuations and any natural cyclical tendencies of the system determine the variation in the output stream

for this run. In the second type, called a signal run, we dynamically change the input factors' settings between specified high and low values during the course of the simulation run. If none of the factors have an appreciable effect on the system, then the spectrum of the signal run output has the same shape as that of the noise run output. However, if the system response is sensitive to the value of an oscillated factor, then the signal spectrum will exhibit a much higher value (or spike) at the corresponding frequency.

More formally, during the signal run the factor levels are sinusoidally oscillated at distinct frequencies ($\omega_i$, $i$ = 1,2,…,$k$) referred to as driving frequencies. Each input term $x_i$ has an associated set of term indicator frequencies in the output. For example, if frequencies $\omega_1 = 0.1$ and $\omega_2 = 0.25$ (expressed in cycles per observation) are assigned to the two input factors $x_1$ and $x_2$, respectively, then the set of frequencies {0.1}, {0.1 ± 0.25}, and {0.1 × 2} are the term indicator frequencies for $x_1$, the $x_1x_2$ interaction, and $x_1^2$, respectively. Trigonometric relations determine the set of term indicator frequencies, which all fall between 0 and 0.5 cycles per observation. Careless choice of factor driving frequencies can partially confound term indicator frequencies, but a simple search algorithm will generate unconfounded designs. Jacobson et al. (1991) provide a table of designs for second-order polynomials with up to 21 factors, as well as for third-order polynomials with up to eleven factors, and mark the designs that are known to maximize the minimum spacing between indicator frequencies (also known as the bandwidth). The 'design' program described in the Appendix is an implementation of their driving frequency selection algorithm that allows one to determine unconfounded designs for second-order models with an arbitrary number of factors.

The values of continuous input factors in the signal run are sinusoidally oscillated:

$$x_i(t) = 0.5(U_i + L_i) + 0.5(U_i - L_i)\cos(2\pi\omega_i(t)), \tag{1}$$

where $x_i(t)$ is the value of the continuous factor $x_i$ at the simulated time index $t$, $U_i$ and $L_i$ are the upper and lower bounds of the factor range, and $\omega_i$ is the driving frequency in cycles per observation. The quantities $0.5(U_i + L_i)$ and $0.5(U_i - L_i)$ are the nominal value and the amplitude of the factor $x_i$, respectively.

For a binary input factor $x_i$ that assumes discrete values $a_1$ or $a_2$, one cannot sinusoidally oscillate the factor level itself. Instead, we can oscillate the probability as follows:

$$\Pr(x_i(t) = a_1) = 0.5 + 0.5\cos(2\pi\omega_i(t)) \tag{2}$$

where $\omega_i(t)$ is the driving frequency for $x_i$ evaluated at time $t$ (Sanchez and Sanchez, 1991).

To facilitate comparison of cyclic behavior, we change from the time domain to the frequency domain by computing Fourier spectra for the signal and noise runs. Let $f_N^2$ denote the spectrum estimator for the noise (or control) run, and let $f_S^2$ denote the spectrum estimator for the signal run. The spectral signal-to-noise (S/N) ratio $f_S^2 / f_N^2$ (or its logarithm) is the basis of the analysis.

The graph of the S/N ratio versus $\omega$ will show a spike for each important term at its term indicator frequencies. Formal analysis of the spectra is also possible. Since the distribution of the sample spectrum is asymptotically proportional to a chi-squared distribution, the distribution of the sample S/N ratio is approximately an $F$ distribution with degrees of freedom dependent on the method used to calculate the spectra. Table 1 summarizes the FDE procedure.

*** Table 1 about here ***

Table 1 refers to the programs provided in the Appendix in many steps. These programs use reasonable defaults in estimating the spectra. Note that calculating spectral terms at exactly $d$ non-zero frequencies (if $d$ is odd) or $d/2$ non-zero frequencies (if $d$ is even) means that each indicator frequency will be associated with a unique S/N ratio. Users who wish to fine-tune their analyses can do so by adjusting parameters such as run length, window type (e.g., Tukey, Parzen, or truncated) and window size $M$. View the source code and a time-series text such as Chatfield (2004) for more information.

## 4. EXAMPLES

We illustrate the FDE methodology for two experiments. First, in Section 4.1, we examine a mathematical simulation where the 'true' metamodel is known but obscured by random noise. Our purpose is to show that the FDE methodology correctly identifies the system factors. In Section 4.2 we examine the use of FDE for screening purposes for a system that is more representative of those for which OM researchers might employ the technique. We simulate a JIT system with kanban in which 34 factors are of interest to the analyst, and quadratic effects, interstage- and intrastage-interactions potentially impact system performance. Note that these examples are primarily to illustrate the methodology, rather than to gain insights into the specific systems.

### 4.1. KNOWN STOCHASTIC SYSTEM

Suppose three factors are of interest: $x_1$, $x_2$, and $x_3$. We generate a time series from the following underlying relationship:

$$Y(t) = 3x_1(t) - x_3(t) + 2x_1(t)\,x_3(t) - x_2^2(t) + \varepsilon_t \tag{3}$$

where the $\varepsilon_t$ are standard normal random variables from a first-order autoregressive process with autocorrelation $\rho = 0.5$.

*Step 1*. Our performance measure is the simulation output, *Y*.

*Step 2*. Three factors ($x_i$, $i$ = 1,2,3) will be oscillated during the signal runs to assess their impact on *Y*.

*Step 3*. All factors will be oscillated between -1 and +1.

*Step 4*. The three driving frequencies are {1/27}, {4/27}, and {10/27}. These are also the indicator frequencies for the main effects. Indicator frequencies corresponding to the three two-way interactions are {3/27, 5/27}, {9/27, 11/27}, and {6/27, 13/27}. Indicator frequencies for the quadratic effects are {2/27}, {8/27}, and {7/27}.

*Steps 5* and *6*. Table 2 shows output from the 'design' program. The first part of this table shows the assigned driving frequencies for each of the three signal runs. The second part shows information useful for analysis, i.e., the frequencies at which main, interaction, and quadratic effects would appear. The notation for the terms is i:0 for the main effect of $x_i$, i:j for an $x_i x_j$ interaction, and i:i for the quadratic $x_i^2$. Since the design is unconfounded, each indicator frequency is associated with at most one term in a run.

*** Table 2 about here ***

*Step 7*. The default run length is $N \geq (\nu + 1)d$ where $\nu \geq 10$. This allows one complete cycle to be truncated to remove any initialization bias, and keeps $\nu$ complete cycles for analysis purposes.

*Step 8*. Sample results from 100 observations of the signal and noise runs appear in Figure 4(a-b). The noise run shows the natural variability in output as a function of time. The strong oscillatory behavior in the signal run clearly indicates that variation in the factors is transmitted to the response.

*** Figure 4 about here ***

*Step 9*. We discard the first full cycle of data from all runs, leaving 270 observations in each run.

*Step 10*. We use the 'fspect' program in the Appendix (with the default "Tukey" window, 27 non-zero frequencies, and the recommended window size *M*=(8/3)27=72) to compute the Fourier spectra of the signal and noise runs. Spectra for the first pair of runs are provided in Figure 5.

*** Figure 5 about here ***

*Step 11.* The 'ratio' program in the Appendix Figure 5 calculates the signal-to-noise (S/N) spectral ratios for each frequency assignment.

*Step 12*. The 'analyze' program pools the results by term across the three frequency assignments and provides a single table (Table 3) with the pooled signal-to-noise ratios. Pooling terms at non-indicator frequencies also provides a lack-of-fit indicator that can be used to determine whether or not a second-order model is sufficient.

*** Table 3 about here ***

*Step 13*. The pooled S/N results (Table 3) clearly indicate spikes (important effects) at frequencies that correspond to the terms $x_1$, $x_3$, the $x_1x_3$ interaction, and the quadratic term $x_2^2$. The values for the lack-of-fit indicator and the other potential model terms (quadratics for $x_1$ and $x_3$, as well as interactions involving $x_2$) are all near one, indicating that this second-order model is sufficient and higher-order effects are not present. FDE has correctly identified the important terms in equation (3).

### *4.2. A KANBAN EXAMPLE*

We now analyze part of a manufacturing plant with three serial stages, and assume a single-card constant-cycle kanban system is used with a one-to-one container relationship. For simplicity and clarity of presentation, the system produces only one item. Figure 6 shows the schematic of the model. While the system's behavior is easy to analyze in the deterministic case, our simulation introduces randomness into the lead time for input materials, machine operating and repair times, machine operating durations, setup times, and demand. We now illustrate the noise factor screening process for the kanban system of Figure 6. Our purpose is to demonstrate, using a problem of interest in operations management, that it is possible to examine many factors, interactions and quadratic effects simultaneously. As with virtually all DOE scenarios, our results are specific to this particular configuration and should not be generalized to kanban systems as a whole.

*** Figure 6 about here ***

*Step 1*. Our performance measure is the service level, measured in terms of average daily number of backorders. An ideal JIT system satisfies all demand on time.

*Step 2.* We investigate 34 different environmental noise factors in six categories: demand volume, processing time, setup time, time between breakdowns, repair time, and supplier lead time. Of the six categories, demand volume and supplier lead time represent external noise factors, and may be the most difficult to control. The remaining four categories contain internal noise factors. The demand category consists of two factors: the mean and variance of the demand volume imposed at stage 1. We assume demand follows a truncated normal distribution. Similarly, the mean and variance are the two factors in the supplier lead time category, which directly impacts stage 3. The setup time category consists of a single noise factor (mean setup time) for each production stage. The range associated with the mean setup time reflects the expected future reduction in setup time due to enhancement programs that will be undertaken by the company. The processing time, time between breakdowns, and repair time categories each consist of three variables per stage: the distribution (truncated normal or exponential), the mean, and the variance corresponding to the truncated normal distribution.

*Step 3.* The factor levels appear in Table 4. We choose factor levels that are generally compatible with the values used in Krajewski et al. (1987) in that they fall between the so-called U.S. Low (a favorable environment for kanban implementation) and Kanban High (an unfavorable environment). We have added a few variables in order to include the distributional shape and variance in many of the categories. The value in Column 3 corresponds to the level for which the noise experiments were conducted, while Columns 4 shows the lower and upper factor levels for the signal runs.

*** Table 4 about here ***

We perform our analysis for a kanban system with fixed decision variables. The analog for a practitioner would be a factor-screening experiment to identify how noise factors impact the performance of a specific kanban system—such as the one currently in use. We set the container size to 10 units and the kanban review period to 480 minutes, so a detached kanban becomes a production order at the beginning of the next day. We then set the number of kanbans to 15 using the procedure of Moeeni and Chang (1990).

*Step 4.* The 'design' program generates the set of 34 driving frequencies. These have the form $\omega_i = f_i / 7656$ ( $i$=1,…,34).

*Step 5* and *6*. The last three columns of Table 2 provide the three driving frequency assignments, denoted by A1, A2, and A3. The discrete factors are assigned to the nine lowest frequencies.

*Step 7*. We use the minimum default run length of $N$=84,216 days.

*Steps 8* and *9*. We conducted the experiments on a Mac PowerBook G4. Our simulation automatically truncated the first cycle (7,656 days) within each run.

*Step 10*. The 'fspect' program in the Appendix is used to compute Fourier spectra. The default Tukey window is used, and spectral terms for 3,828 non-zero frequencies are printed.

*Step 11*. The 'ratio' program computes the signal-to-noise spectral ratios for the three different frequency assignments. These are illustrated in Figure 7(a-c). Since indicator frequencies differ across the three driving frequency assignments, we identify the major spikes on each subgraph according to the factor term.

*** Figure 7 about here ***

*Step 12*. The 'analyze' program pools the S/N ratios for term indicator frequencies across like terms, resulting in a total of 629 S/N ratios. These correspond to 34 main effects, 34 quadratic effects, and 561 two-factor interactions.

*Step 13*. In Table 3, we list all term identifiers that result in an average S/N ratio (across the three driving frequency assignments) of at least 10.00. While somewhat arbitrary, this cutoff is well above both the background levels and the average for the remaining indicator frequencies. Table 5 lists quadratic, intra-stage, and inter-stage interactions in addition to main effects.

*** Table 5 about here ***

## 5. COMPARING COMPUTATIONAL REQUIREMENTS

FDEs (like other experimental designs) are more efficient than trial-and-error approaches to identify the extent of factors' impacts on simulation performance. The orthogonality of FDEs also eliminates problems of multicollinearity among input factors, which can make it more difficult to identify statistically significant terms. A natural question to ask is how this method compares (in terms of implementation effort) to other orthogonal experimental designs.

We find the oversight required for FDE is much less than that for run-oriented designs involving even a moderate number of factors. While one must write (or modify) the simulation code to accept time-varying inputs, implementing a FDE (using the programs in the Appendix) and gathering the results is then an almost fully-automated process. As such, it is less prone to data entry and data collation errors than is an experiment requiring, say, $2^{10}$=1,024 runs at distinct configurations, unless programs or scripts are developed to automate the data generation and collection process.

The amount of data required is another characteristic that has been used to compare alternative experimental designs. In what follows we say one design is more efficient than another if it can estimate the same effects with less data. Clearly, resolution III fractional factorials and other so-called screening designs require very few runs. However, we do not consider these to be direct competitors for FDEs since they do not allow the analyst to test for the existence of quadratic effects or two-way interactions. We recommend the use of FDEs for screening purposes when the analyst seeks to identify important *terms in a second-order model*, rather than solely important *main effects*.

This means that the data requirements of FDEs should be compared to those of other orthogonal designs that permit tests of all main effects, quadratic effects, and two-way interactions. Full factorials are candidates, but these are notoriously inefficient when higher-order interactions are assumed negligible. For example, a $2^{34}$ factorial experiment would require 17.2 billion computer runs to estimate only 629 terms—even if each run took only one CPU second it would take over 544 years of CPU time to finish the experiment! A resolution V central composite design ($CCD_V$) is considered an efficient design for second-order response models. It is convenient to discuss CCDs in their coded levels, where each factor ranges from a low of -1 to a high of +1. $CCD_V$ designs involving $k$ factors are composed of

- A $2^k$ factorial or $2^{k-p}$ fractional factorial design of resolution V or higher;
- $k$ additional pairs of *star points*: factor $i$ takes the value +a or -a in the $i$th pair of design points (a=1 is possible), while all other factors are set to zero (the middle level);
- One or more center points at the design point {0,0,…,0}. We use 2 center points.

Fractional factorials and CCDs appear in texts such as Box et al. (1978) or Montgomery (2000). Kleijnen et al. (2005) discuss the use of these and other designs for simulation experiments. The National Institute of Science and Technology has a nice description on their website (NIST/SEMATECH 2005), and some statistical software packages also include experimental design options. Still, resolution V designs are only presented for a relatively small number of factors: NIST/SEMATCH (2005) show $k$=8, while Box et al. tabulate a $2_V^{11-3}$ fractional factorial that could be used as the basis of a $CCD_V$ involving 11 factors. With the lack of published designs allowing full estimation of second-order models of the response for a larger number of factors, assessing the relative computational requirements of FDE seems problematic.

However, a discrete orthogonal basis set called Walsh sequencies can be used to generate two-level designs. Sanchez et al. (2001) show that by oscillating factors at carefully chosen Walsh sequencies, the resulting design points are simply those corresponding to a full factorial design (reordered). So, rather than specifying a factorial design in

terms of a $2^k \times k$ design matrix, it can be specified by an assignment of the $k$ factors to $k$ Walsh sequencies. Sanchez and Sanchez (2005) use this idea, together with a simple iterative algorithm, to generate highly efficient resolution V fractional factorials. We now use these to construct extremely efficient CCDs as a basis for assessing the efficiency of FDEs involving up to 120 factors.

The first few columns of Table 6 list the denominator for the FDE frequency assignments ($d$) and the number of design points in a single replication of the efficient CCD ($c$). The remaining columns require some explanation. A single simulation run (e.g., for a queueing system simulation) may yield an output stream where the data are correlated so standard statistical techniques cannot be used directly. A common way of dealing with this phenomenon is the *method of batch means* (Law and Kelton, 2000). The output stream is split into batches large enough so that observations more than one batch apart are effectively independent. After discarding the initial batch to remove any warm-up period, the batch means are treated as i.i.d. observations for analysis purposes. This means that a long run may be required to gain a single number (batch mean) using a run-oriented approach. This must be taken into account when computing data requirements for run-oriented designs.

*** Table 6 about here ***

We consider several systems with different output correlation behavior in Table 6, such as first-order autoregressive models with differing levels of lag 1 autocorrelation. These are denoted by AR($\rho$). Autocorrelations of $\rho = 0.5$ to to $\rho = 0.995$ correspond to (theoretical) batch sizes of 5 to 598, where the batch size is the value $t$ when the theoretical lag $t$ autocorrelation first drops below 0.05. Two other designs indicate batch sizes corresponding to the noise run of our kanban system ($t = 3000$) and a pseudo-system corresponding to a queue with heavy traffic intensity ($t = 5000$). The minimum data required for a single replication of a CCD is then $N_{CCD} = 2ct$ (if we assume the analyst runs only two complete batches for each design point and discards the first). The total data requirement for one of our FDE experiments is $N_{FDE} = 66d$, corresponding to three sets of signal and noise runs, each involving 11 complete cycles (again, discarding the first). The efficiencies reported in the table are $N_{FDE} / N_{CCD}$. Note that while CCDs are more efficient for independent samples or low correlations, FDEs require substantially less data when the systems are highly correlated. For example, for the kanban model the FDE needed only 4% of the data that would have been required for a single replication of a CCD involving 34 factors. Viewed another way, this 34-factor FDE requires about the same amount of data as an 8-factor CCD.

To compute the relative efficiency of our FDEs to CCDs where *b* usable batches are obtained from each run, divide the entries in Table 6 by ($b$+1)/2. A rule of thumb in simulation output analysis is to use between 8 and 20 batches, and analysts often use a common batch size estimated from a run they expect to have the high autocorrelation, such as the heaviest traffic conditions for a queueing simulation. If this advice is followed, FDEs may require less total data than CCDs even when the system output exhibits only moderate correlation.

Since analysts have not had ready access to large resolution V designs, how have they been conducting experiments involving many factors? Often, screening experiments (which test only for main effects) are used to save time and data collection effort in the initial stages of exploration/experimentation. Based on the screening results, one can conduct more detailed experiments using a small subset of the original factors and interaction terms. For example, an analyst might use a saturated fractional factorial experiment involving 15 factors to identify factors with important main effects. If four of these are found to be significant, a $3^4$ factorial might then be performed to check for interactions and quadratic effects. In contrast, conducting an FDE that examines quadratic and interaction effects requires no more simulation runs than an FDE that examines only main effects. Spikes at non-indicator frequencies show the lack-of-fit of an under-specified model. If the simulation output exhibits high autocorrelation, FDE may still require fewer total observations than a run-oriented design. Thus, FDE can often provide more insight into the simulation's behavior for the same computational effort, because it does not require one to make dangerous assumptions of less complex model structures.

When using the programs in the Appendix, FDE has one additional advantage—the results (e.g., Table 2) are ready for interpretation. In contrast, even if an analyst conducting a run-oriented design automates the data generation and collection process, they typically port the output into a statistical software package and must spend additional time in order to conduct the analysis. Consider the kanban example of Section 4.2. Although the signal and noise runs were relatively long, the computer experiments required only 50 seconds of CPU time on a Mac PowerBook G4. The calculation of the Fourier spectra took under three minutes, and that for the spectral ratios was negligible.

## 6. SOME DETAILS, CAVEATS, AND REFINEMENTS

Our presentation of FDE and our implementation via the programs in the Appendix is intended to provide a reader with a general-purpose analysis approach. Nonetheless, there are some details and caveats worth mentioning. While we have presented FDEs as a useful tool for simulation experiments, the methodology is not a panacea.

First, some details. In this paper we use a qualitative approach to identify important terms in the model. This is reasonable since the magnitude of a spike is proportional to the square of the magnitude of its effect, but formal tests are also possible since the pooled S/N ratios asymptotically follow an $F_{\nu,\nu}$ distribution (Sanchez and Buss, 1987).

In our description of the oscillation patterns, we distinguish between quantitative and binary factors. Clearly, some simulations may involve qualitative factors with 3 or more levels, or discrete factors with a limited number of potential choices. These cases are discussed in Sanchez and Sanchez (1991). The former can be dealt with by constructing a tree with binary factors associated with each split. This increases the number of factors that will be oscillated during the experiment and input for the 'design' program to generate frequency assignments. Binary trees are also options for discrete quantitative factors; an alternative is to round the oscillated values to the nearest integers.

We specify three sets of frequency assignments for an FDE. However, if the noise spectrum is flat, a single pair of signal and noise runs suffices. An analyst faced with very long run times may wish to begin by examining the spectrum of a single noise run to assess whether or not all three assignments are necessary.

A few caveats are in also in order. For many queueing networks, such as the kanban system of this paper, certain configurations may make the system unstable. For example, under sufficiently high traffic intensity, backorders can build up infinitely. If this is true, the output stream will 'drift' in terms of its mean value. The corresponding result in the spectrum is extreme low frequency behavior (i.e., a large spike at frequency zero). If this occurs, one should take care in interpreting the results. Morrice and Jacobson (1995) have begun to address the use of FDE for transient models, but their work deals with small oscillation amplitudes rather than the large amplitudes used in this example. We remark that unstable system configurations are also a potential problem with run-oriented approaches.

Fourier spectral computations typically use window estimators to estimate the spectral terms. At times, the effects due to an important factor may be 'smeared' across adjacent terms. This may be observed visually if there are many 'hills' rather than 'spikes' in the S/N ratio plot. Our experience has been that the defaults work well, but if smearing appears to be a problem, one can rerun the 'fspect' program on the simulation output, while specifying a larger window size as part of the command line input. Doubling the window size from the default value halves the degrees of freedom associated with the pooled S/N ratios (intermediate values can also be used). If desired—and if time permits—longer runs or additional pairs of signal and noise runs can be made to allow larger window sizes to be used without sacrificing degrees of freedom.

Before calculating the spectra, we can check to determine if truncating a single cycle is sufficient by calculating the autocovariance terms of the simulation output. A rule of thumb (Law and Kelton, 2000) states that observations separated by at least 10 times the lag at which the magnitude of the autocorrelation last drops below 0.4 are approximately independent. This is a more stable calculation than seeking the time when the magnitude of the autocorrelation last drops below 0.05 when the autocorrelations are themselves estimates.

Finally, our comparisons between FDEs and CCDs (in Section 5) dealt with the data required to conduct experiments, rather than other measures such as the power of the resulting *F*-tests for identifying significant terms or the precision of the fitted metamodel coefficients. A broad investigation is beyond the scope of this paper, but Sanchez and Konana (2000) empirically investigate FDE's ability to correctly identify model terms for a suite of AR(1) systems.

For the interested reader, we briefly describe some refinements and extensions of the basic FDE approach. Morrice and Schruben (1993) proposed a variation of FDE called harmonic analysis. Here the simulation output is regressed onto explanatory variables of the form $\sin(2\pi\omega_i(t))$ and $\cos(2\pi\omega_i(t))$ for all indicator frequencies $\omega_i$. This allows one to obtain $R^2$ values and metamodel coefficients, but at the costs of an extremely large analysis matrix without lack-of-fit indications. However, this may be a useful second-stage analysis after FDEs have identified an appropriate model.

Sanchez and Konana (2000) examined the effects of different run lengths for the signal and noise runs. Their results show that it is often possible to obtain the same screening power with fewer total observations by allocating more observations to the signal run. In this context, Sanchez and Konana (2000) also examined the use of spectral differences (Sargent and Som, 1992; Robinson et al., 1993) rather than spectral ratios.

One can use a variation of the frequency domain approach to examine system robustness. For example, Moeeni et al. (1997) use response-surface methodology to model the mean and variability of service and inventory levels for the kanban system of Section 4.2 as a function of a common container size, the number of kanbans, and the kanban lead times, where the mean and variability are expectations over the ranges of the 34 noise factors of Table 2. In this instance, using nearly-saturated fractional factorial designs for the noise factors would still result in 64 configurations for each combination of the decision factors. Since the estimation of specific noise factor effects is not the goal in this study, the FDE methodology is efficient but not restrictive. Sanchez and Wu (2003) developed and demonstrated a frequency-based approach for terminating simulations.

## 7. CONCLUDING REMARKS

We have shown that frequency domain experimentation makes it possible to simultaneously analyze many factors in a single (albeit, large) experiment. Our goal is to make this approach accessible to researchers and practitioners in operations management. We have synthesized results appearing in several archival research journals to present a step-by-step approach for simulation experiments using frequency domain experimentation. Our hope is that this methodology may enrich the theory and practice of operations management by providing researchers and analysts a tool to help identify factors that impact their production or service systems.

While the methods for FDE have been published, they are not yet widely known. Many academic researchers using simulation face the prospect of designing experiments to analyze a complex system or problem. For one-time applications, the prospect of writing programs to choose appropriate driving frequencies, calculate sample autocovariances, compute Fourier spectra and combine spectral ratio terms appropriately for multiple frequency assignments may have been too daunting. We feel this paper shows the benefits of simultaneous analysis, and by providing access to the programs needed for FDE design and analysis we hope to facilitate the technique's use.

FDE is a heuristic method for identifying important model terms. This may be only a preliminary step in addressing a research question, but it is an important one. Many studies have focused on main effects due to time limitations, yet it is clear that complex systems cannot typically be characterized by simple, main-effect models. We have shown that FDEs are very efficient designs for exploring simulations with correlated output. FDEs can help the analyst determine which factors impact a system, and hence require oversight. Alternatively, they can identify terms for further experiments or analysis using conventional methods.

**APPENDIX**

A suite of tools to support frequency domain experiments, as well as the sample model of Section 4.2, is available in source form on request from the authors. A README.txt file describes the components ('design', 'fspect', 'ratio' and 'analyze'), as well as instructions for running the provided example.



**ACKNOWLEDGMENTS**

Susan M. Sanchez was supported in part by the National Research Council and grants from the United States Marine Corps.

**REFERENCES**

Box, G.E.P., W.G. Hunter, and J.S. Hunter, 1978, Statistics for experimenters: an introduction to design, data analysis and model building (Wiley, New York).

Cabrera-Rios, M., C.A. Mount-Campbell, and S.A. Irani, 2002, An approach to the design of a manufacturing cell under economic considerations, International Journal of Production Economics 78, 223-237.

Chatfield, C., 2004, The analysis of time series: an introduction, 6th edition (CRC Press LLC, Boca Raton, Florida).

Enns, S.T., 1995, An integrated system for controlling work load and flow in a job shop, International Journal of Production Research 33, 2801-2820.

Jacobson, S.H., A.H. Buss, and L.W. Schruben, 1991, Driving frequency selection for frequency domain simulation experiments, Operations Research 39, 917-924.

Jensen, J.B., M.K. Malhotra, and P.R. Philipoom, 1996, Machine dedication and process flexibility in a group technology environment, Journal of Operations Management 14, 19-39.

Kim, S.-C. and P.M. Bobrowski, 1995, Evaluating order release mechanisms in a job-shop with sequence-dependent setup times, Production & Operations Management 4, 165-180.

Klassen, K.J. and T.R. Rohleder, 1996, Scheduling outpatient appointments in a dynamic environment, Journal of Operations Management 14, 83-101.

Kleijnen, J.P.C., 1993, Simulation and optimization in production planning: a case study, Decision Support Systems 9, 269-280.

Kleijnen, J.P.C., S.M. Sanchez, T.W. Lucas, and T.M. Cioppa, 2005, A user's guide to the brave new world of simulation experiments (with online supplement), INFORMS Journal on Computing 17, 263-269.

Krajewski, L.J., B.E. King, L.P. Ritzman, and D.S. Wong, 1987, Kanban, MRP, and shaping the manufacturing environment, Management Science 33, 39-57.

Law, A. M. and W.D. Kelton, 2000, Simulation Modeling and Analysis, 3rd edition (McGraw-Hill, New York).

Malhotra, M.K. and L.P. Ritzman, 1994, Scheduling flexibility in the service sector: a postal case study, Production & Operations Management 3, 100-117.

Moeeni, F. and Y.L. Chang, 1990, An approximate solution to deterministic kanban systems, Decision Sciences 21, 596-607.

Moeeni, F., S.M. Sanchez, and A.J. Vakharia, 1997, A robust design framework for just-in-time systems with kanban, International Journal of Production Research 35, 2821-2838.

Montgomery, D. C., 2000, Design and analysis of experiments, 5th edition (Wiley, New York).

Morrice, D.J. and S.H. Jacobson, 1995, Amplitude selection in transient sensitivity analysis, in: C. Alexopoulos, K. Kang, W.R. Lilegdon, and D. Goldsman, eds., Proceedings of the 1995 Winter Simulation Conference (Institute of Electrical and Electronic Engineers, Piscataway, New Jersey) 330-335.

Morrice, D.J. and L.W. Schruben, 1993, Simulation factor screening using harmonic analysis, Management Science 39, 1459-1476.

NIST/SEMATECH, 2005, e-Handbook of Statistical Methods, http://www.itl.nist.gov/div898/handbook/ (accessed online on March 25, 2005).

Robinson, J.K., L.W. Schruben, and J.W. Fowler, 1993, Experimentation with large-scale semiconductor simulations: a frequency domain approach to factor screening, in: D. Mitta, L. Burke, G. Tonkay, J. English, J. Galliamore, and G. Klutke, eds., Proceedings of the Institute of Industrial Engineers Research Conference (Institute of Electrical and Electronic Engineers, Piscataway, New Jersey) 112-116.

Sanchez, P.J. and A.H. Buss, 1987, A model for frequency domain experiments, in: A. Thesen, H. Grant, W.D. Kelton, eds., Proceedings of the 1987 Winter Simulation Conference (Institute of Electrical and Electronic Engineers, Piscataway, New Jersey) 424-427.

Sanchez, P.J. and S.M. Sanchez, 1991, Design of frequency domain experiments for discrete-valued factors, Applied Mathematics and Computation 42, 1-21.

Sanchez, P.J., K.L. Head, and J.S. Ramberg, 2001, Life in the fast lane: Yates' algorithm, fast Fourier and Walsh transforms, in: M. Dror, P. L'Ecuyer, F. Szidarovszky, eds., Modeling Uncertainty (Kluwer Academic Publishers, Norwell, Massachusetts) 652-684.

Sanchez, S.M. and P. Konana, 2000, Efficient data allocation for frequency domain experimentation, Operations Research Letters 26, 81-89.

Sanchez, S.M. and P.J. Sanchez, 2005, Very large fractional factorial and central composite designs, working paper, Operations Research Department, Naval Postgraduate School, Monterey, California.

Sanchez, S.M. and H.-F. Wu, 2003, Frequency-based designs for terminating simulation experiments: a peace-enforcement example, In: S. Chick, P.J. Sánchez, D. Ferrin, and D. Morrice, eds., Proceedings of the 2003 Winter Simulation Conference (Institute of Electrical and Electronic Engineers, Piscataway, New Jersey) 952-959.

Sargent, R.G. and T.K. Som, 1992, Current issues in frequency domain experimentation, Management Science 38, 667-687.

Schruben, L.W. and V.J. Cogliano, 1987, An experimental procedure for simulation response surface model identification, Communications of the ACM 30, 716-730.

Vakharia, A.J., D. Parmenter, and S.M. Sanchez, 1996, The operating impact of parts commonality, Journal of Operations Management 14, 3-18.

**List of figure captions**

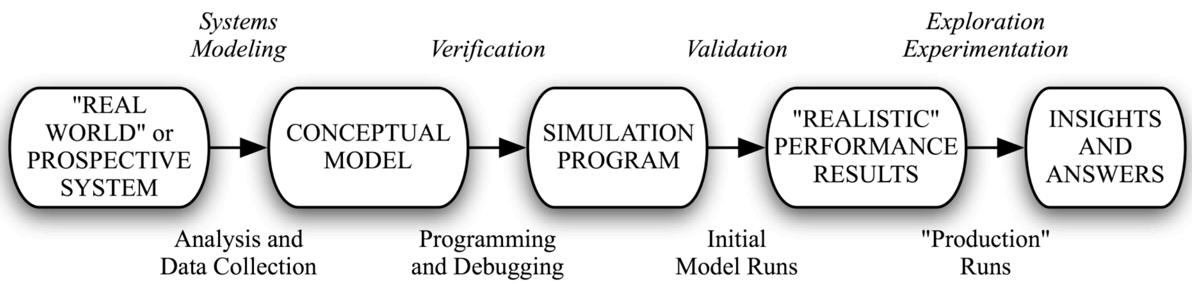



Fig 1. Simulation overview

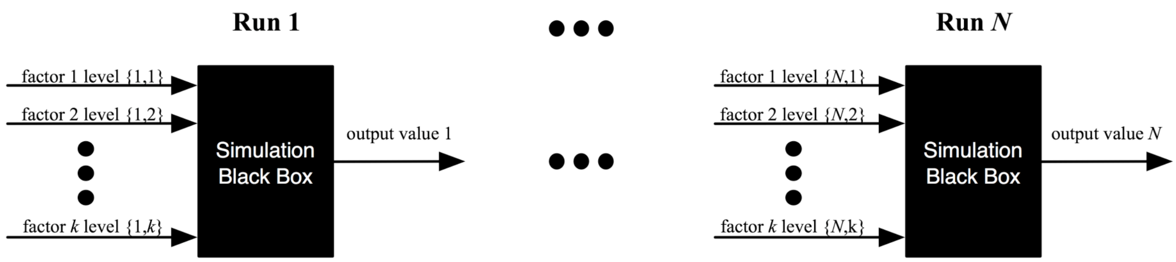



Fig. 2.  Run-oriented experiments

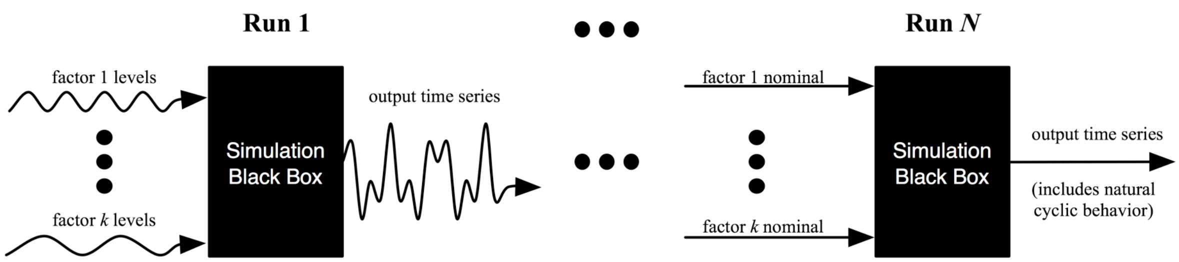
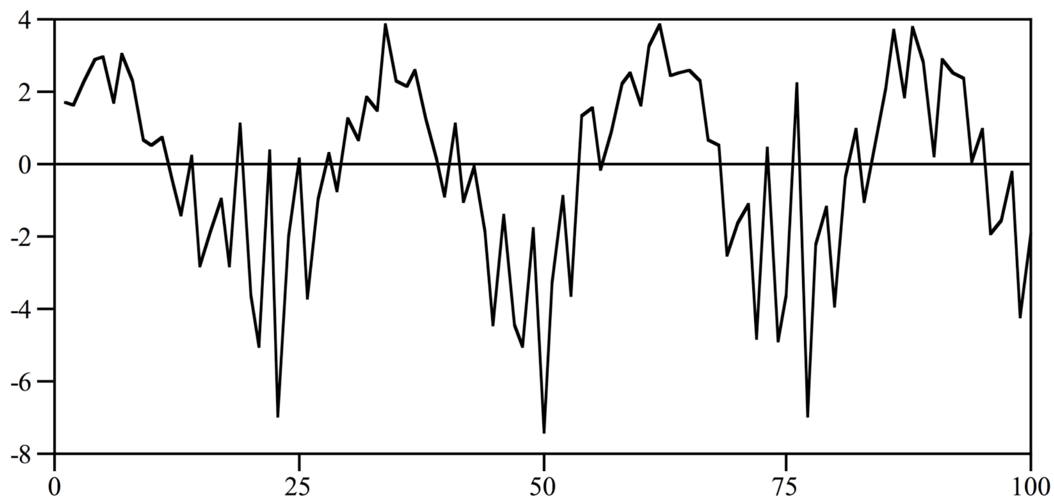
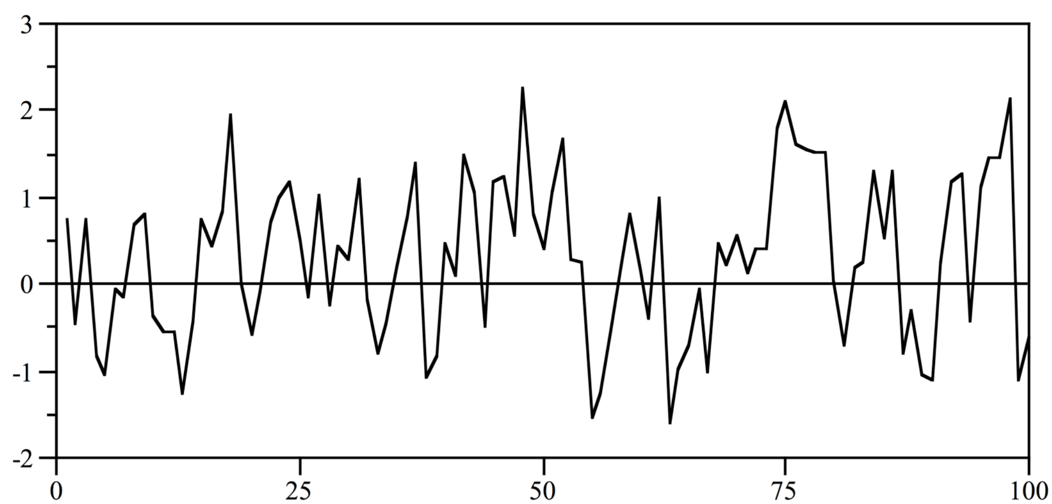



Fig. 3. Frequency domain experiments

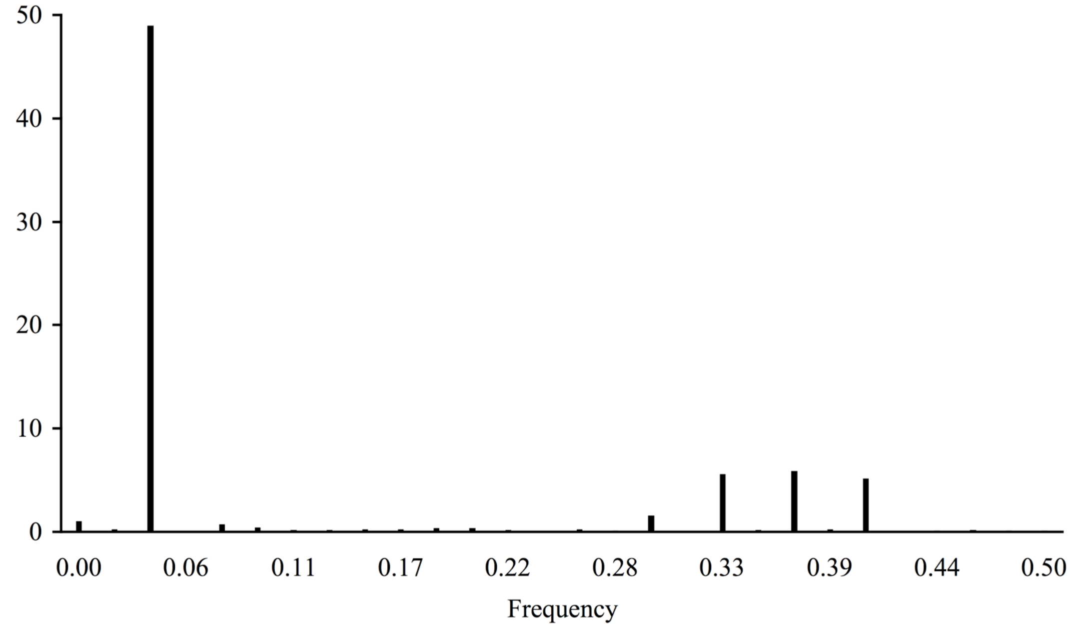



(a) Signal run (first 100 observations)

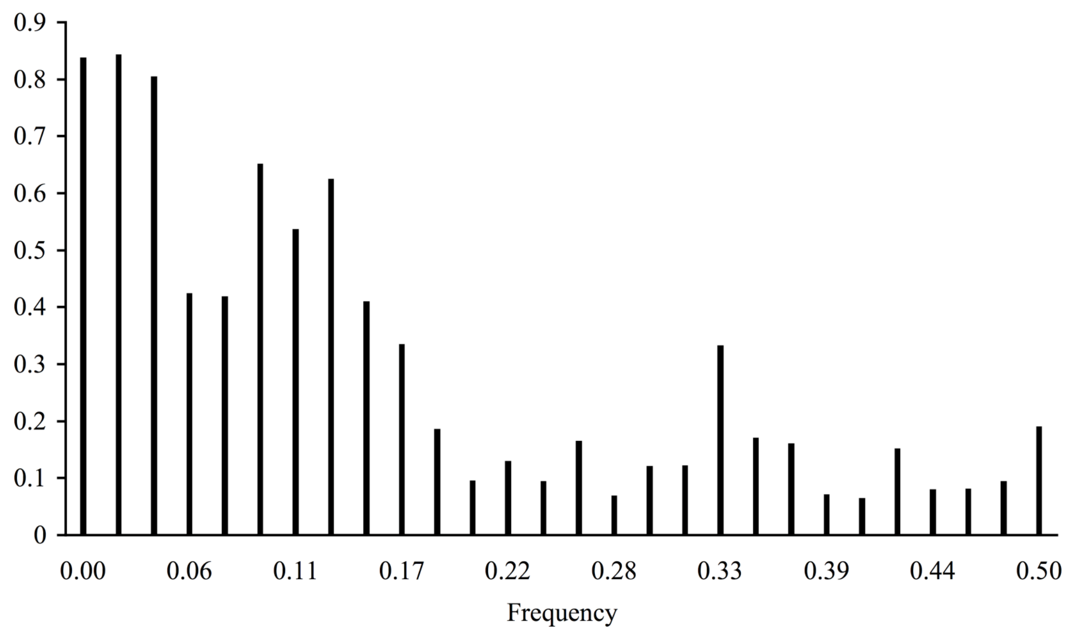



(b) Noise run (first 100 observations)

Fig. 4. Sample output for mathematical example




(a) Signal run




(b) Noise run

Fig. 5. Spectra for mathematical example

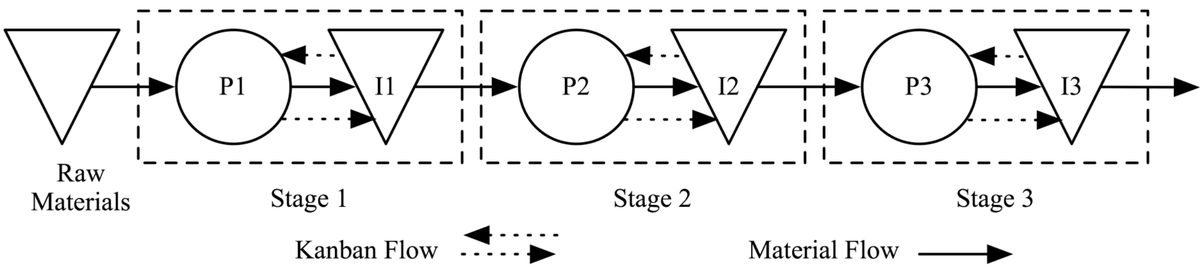
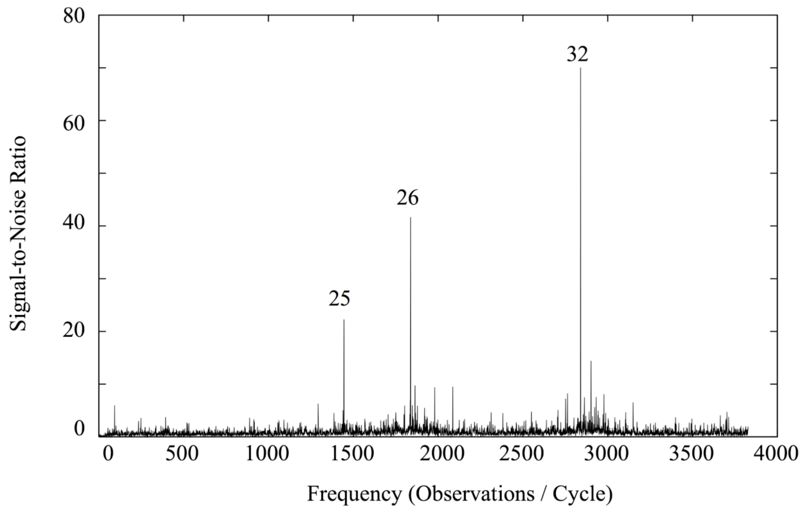
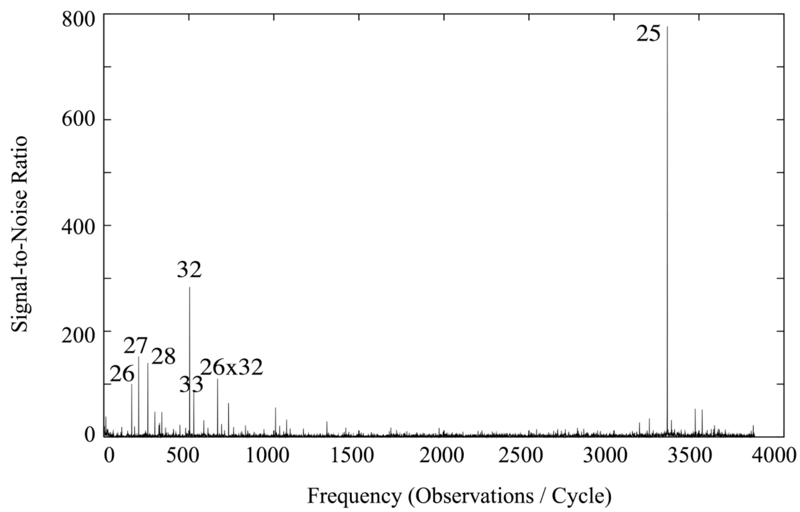


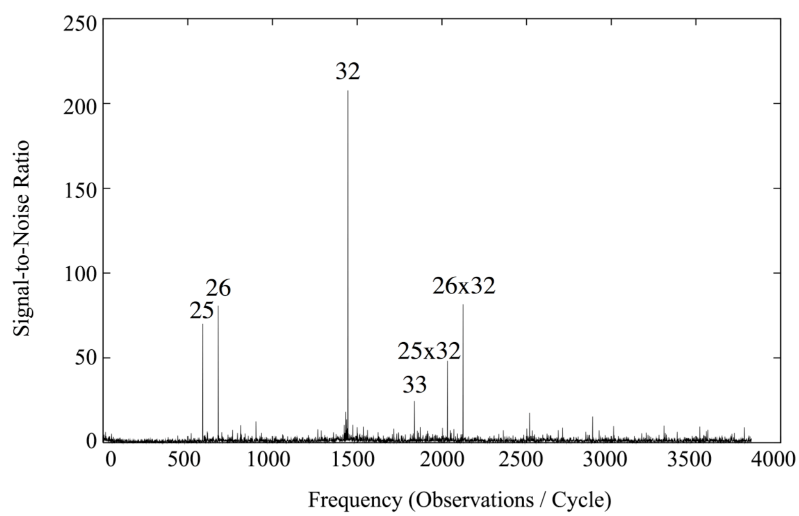

Fig. 6. Schematic of a kanban system




(a) Frequency assignment 1




(b) Frequency assignment 2




(c) Frequency assignment 3

Fig. 7. Signal-to-noise spectral ratios for kanban example

Table 1
FDE Summary

| Step | | Description |
|---|---|---|
| 1. | Specify performance measures. | Determine output measures that are important for purposes of the analysis. |
| 2. | Specify factors. | Determine the factors to be investigated. |
| 3. | Choose factor ranges. | Set the low and high levels of interest [$L_i$, $U_i$] for each factor *i*. |
| 4. | Identify driving frequencies. | Use the 'design' program of the Appendix to identify a suitable set of driving frequencies $\omega_i$. Let *d* denote the divisor for the frequency assignment. |
| 5. | Assign driving frequencies to factors to create base design for signal run. | Assign each factor a frequency from step 4. |
| 6. | Reassign driving frequencies to control for system gain. | Generate designs for two additional signal runs by permuting assignments so that each factor is observed at a low, a medium, and a high frequency across the three signal runs. Binary factors should have low (but different) oscillation frequencies across the three signal runs. |
| 7. | Determine run length. | Default choice is $N=(v+1)d$ where $v$ (>= 10) is the desired degrees of freedom in the denominator of the resulting *F* test. |
| 8. | Run experiments. | For the noise runs, each factor is held at its nominal (middle) level. For the signal runs, oscillate each factor as eqn. (1) or (2) using the assignments of Steps 5 and 6. |
| 9. | Truncate output. | Remove any initial bias present by deleting the first complete cycle (*d* observations). |
| 10. | Compute Fourier spectra. | Use the 'fspect' program of the Appendix, specifying *d* non-zero frequencies (*d*/2 if *d* is even) and using a window size of $M = 8d/3$. |
| 11. | Calculate signal-to-noise ratios. | Use the 'ratio' program of the Appendix. |
| 12. | Pool terms. | Use the 'analyze' program of the Appendix to pool the results by term across the three signal-to-noise ratios. |
| 13. | Interpret the results. | Large values (spikes) in the table of pooled results correspond to terms with statistically significant effects, while spikes indistinguishable from background noise indicate no factor effects. |

Table 2
Frequency assignments for the mathematical example

DESIGN

| Factor | Assigned Frequency Run 1 | Run 2 | Run 3 |
|---|---|---|---|
| 1 | 1 / 27 | 10 / 27 | 4 / 27 |
| 2 | 4 / 27 | 1 / 27 | 10 / 27 |
| 3 | 10 / 27 | 4 / 27 | 1 / 27 |

ANALYSIS

| Indicator Frequency Fractional | Decimal | Factors Run1 | Run2 | Run3 |
|---|---|---|---|---|
| 1 / 27 | (0.037037) | 1:0 | 2:0 | 3:0 |
| 2 / 27 | (0.074074) | 1:1 | 2:2 | 3:3 |
| 3 / 27 | (0.111111) | 2:1 | 3:2 | 3:1 |
| 4 / 27 | (0.148148) | 2:0 | 3:0 | 1:0 |
| 5 / 27 | (0.185185) | 2:1 | 3:2 | 3:1 |
| 6 / 27 | (0.222222) | 3:2 | 3:1 | 2:1 |
| 7 / 27 | (0.259259) | 3:3 | 1:1 | 2:2 |
| 8 / 27 | (0.296296) | 2:2 | 3:3 | 1:1 |
| 9 / 27 | (0.333333) | 3:1 | 2:1 | 3:2 |
| 10 / 27 | (0.370370) | 3:0 | 1:0 | 2:0 |
| 11 / 27 | (0.407407) | 3:1 | 2:1 | 3:2 |
| 13 / 27 | (0.481481) | 3:2 | 3:1 | 2:1 |

Table 3
Pooled output for the mathematical example

| | | S/N ratio | # bins pooled |
|---|---|---|---|
| Lack of Fit indicator: | | 1.22 | 45 |
| Factor 1 | Main effect: | 209.74 | 3 |
| | Quadratic effect: | 1.35 | 3 |
| Factor 2 | Main effect: | 1.11 | 3 |
| | Interaction with factor 1: | 1.27 | 6 |
| | Quadratic effect: | 8.76 | 3 |
| Factor 3 | Main effect: | 30.14 | 3 |
| | Interaction with factor 2: | 31.87 | 6 |
| | Interaction with factor 3: | 1.28 | 6 |
| | Quadratic effect: | 1.06 | 3 |

Table 4
Noise Factor Levels and Frequency Assignments for Kanban Example

| Factor Number | Description | Nominal Value | Oscillation Range | Freq. Assignment Numerator* | | |
|---|---|---|---|---|---|---|
| | | | | A1 | A2 | A3 |
| 1 | Repair Time 1 distribution. | Normal | Normal/Exponential | 1 | 17 | 67 |
| 2 | Repair Time 2 distribution. | Normal | Normal/Exponential | 4 | 29 | 89 |
| 3 | Repair Time 3 distribution. | Normal | Normal/Exponential | 10 | 52 | 132 |
| 4 | Breakdown Arrival 1 distribution. | Normal | Normal/Exponential | 17 | 67 | 1 |
| 5 | Breakdown Arrival 2 distribution. | Normal | Normal/Exponential | 29 | 89 | 4 |
| 6 | Breakdown Arrival 3 distribution. | Normal | Normal/Exponential | 52 | 132 | 10 |
| 7 | Processing Time 1 distribution | Normal | Normal/Exponential | 67 | 1 | 17 |
| 8 | Processing Time 2 distribution | Normal | Normal/Exponential | 89 | 4 | 29 |
| 9 | Processing Time 3 distribution | Normal | Normal/Exponential | 132 | 10 | 52 |
| 10 | Raw Material Arrival variance | 5 days$^2$ | 1 – 9 days$^2$ | 164 | 680 | 2086 |
| 11 | Raw Material Arrival mean | 20 days | 10 – 30 days | 205 | 903 | 2117 |
| 12 | Repair Time 1 variance | 2 hours$^2$ | 1 – 3 hours$^2$ | 259 | 1016 | 2195 |
| 13 | Repair Time 2 variance | 2 hours$^2$ | 1 – 3 hours$^2$ | 303 | 1061 | 2613 |
| 14 | Repair Time 3 variance | 2 hours$^2$ | 1 – 3 hours$^2$ | 350 | 1248 | 2840 |
| 15 | Repair Time 1 mean | 8 hours | 7 – 9 hours | 405 | 1358 | 3060 |
| 16 | Repair Time 2 mean | 8 hours | 7 – 9 hours | 505 | 1445 | 3314 |
| 17 | Repair Time 3 mean | 8 hours | 7 – 9 hours | 529 | 1838 | 164 |
| 18 | Breakdown Arrival 1 variance | 164 days$^2$ | 64 – 264 days$^2$ | 588 | 1878 | 205 |
| 19 | Breakdown Arrival 2 variance | 164 days$^2$ | 64 – 264 days$^2$ | 680 | 2086 | 259 |
| 20 | Breakdown Arrival 3 variance | 164 days$^2$ | 64 – 264 days$^2$ | 903 | 2117 | 303 |
| 21 | Breakdown Arrival 1 mean | 120 days | 80 – 160 days | 1016 | 2195 | 350 |
| 22 | Breakdown Arrival 2 mean | 120 days | 80 – 160 days | 1061 | 2613 | 405 |
| 23 | Breakdown Arrival 3 mean | 120 days | 80 – 160 days | 1248 | 2840 | 505 |
| 24 | Demand Volume variance | 34 units$^2$ | 4 – 64 units$^2$ | 1358 | 3060 | 529 |
| 25 | Demand Volume mean | 92 units | 87 – 97 units | 1445 | 3314 | 588 |
| 26 | Setup Time 1 mean | 12 min | 2 – 22 min | 1838 | 164 | 680 |
| 27 | Setup Time 2 mean | 12 min | 2 – 22 min | 1878 | 205 | 903 |
| 28 | Setup Time 3 mean | 12 min | 2 – 22 min | 2086 | 259 | 1016 |
| 29 | Processing Time 1 variance | 0.1 min$^2$ | 0.06 – 0.16 min$^2$ | 2117 | 303 | 1061 |
| 30 | Processing Time 2 variance | 0.1 min$^2$ | 0.06 – 0.16 min$^2$ | 2195 | 350 | 1248 |
| 31 | Processing Time 3 variance | 0.1 min$^2$ | 0.06 – 0 16 min$^2$ | 2613 | 405 | 1358 |
| 32 | Processing Time 1 mean | 3 min | 2 – 4 min | 2840 | 505 | 1445 |
| 33 | Processing Time 2 mean | 3 min | 2 – 4 min | 3060 | 529 | 1838 |
| 34 | Processing Time 3 mean | 3 min | 2 – 4 min | 3314 | 588 | 1878 |

*Denominator is 7,656

Table 5
Average Signal-to-Noise Ratios for Service Performance

| Term Identifier | Description | Average S/N ratios |
|---|---|---|
| 25 | Demand Volume mean | 289.65 |
| 32 | Processing Time 1 mean | 187.21 |
| 26 | Setup Time 1 mean | 74.26 |
| 27 | Setup Time 2 mean | 56.91 |
| 28 | Setup Time 3 mean | 48.40 |
| 26 × 32 | Setup Time 1 mean × Processing Time 1 mean | 42.57 |
| 33 | Processing Time 2 mean | 38.35 |
| $32^2$ | Processing Time 1 $\text{mean}^2$ | 24.41 |
| 27 × 33 | Setup Time 2 mean × Processing Time 2 mean | 16.04 |
| 25 × 26 | Demand Volume mean × Setup Time 1 mean | 15.38 |
| 25 × 32 | Demand Volume mean × Processing Time 1 mean | 14.07 |
| 7 × 25 | Processing Time 1 distn. × Demand Volume mean | 12.88 |
| 25 × 27 | Demand Volume mean × Setup Time 2 mean | 11.70 |
| $26^2$ | Setup Time 1 $\text{mean}^2$ | 11.40 |
| 34 | Processing Time 3 mean | 11.39 |
| 27 × 32 | Setup Time 2 mean × Processing Time 1 mean | 11.25 |
| Other Indicator Frequencies | | 1.68 |
| (Standard Deviation) | | (0.95) |
| Background | | 1.65 |
| (Standard Deviation) | | (1.84) |

Table 6: Relative Efficiency of FDE (6 runs with 11 cycles per run) to a single replication of a CCD for various systems

| | Base designs | | Systems | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Factors | FDE denom ($d$) | CCD runs ($c$) | Indep | AR1(.5) | AR1(.6) | AR1(.7) | AR1(.8) | AR1(.9) | AR1(.95) | AR1(.98) | AR1(.995) | pseudo kanban | pseudo queue |
| 2 | 14 | 10 | 46.2 | 9.2 | 7.7 | 5.1 | 3.30 | 1.59 | 0.78 | 0.310 | 0.077 | 0.015 | 0.009 |
| 4 | 46 | 26 | 58.4 | 11.7 | 9.7 | 6.5 | 4.17 | 2.01 | 0.99 | 0.392 | 0.098 | 0.019 | 0.012 |
| 8 | 221 | 82 | 88.9 | 17.8 | 14.8 | 9.9 | 6.35 | 3.07 | 1.51 | 0.597 | 0.149 | 0.030 | 0.018 |
| 12 | 610 | 156 | 129.0 | 25.8 | 21.5 | 14.3 | 9.22 | 4.45 | 2.19 | 0.866 | 0.216 | 0.043 | 0.026 |
| 23 | 2,367 | 1,074 | 72.7 | 14.5 | 12.1 | 8.1 | 5.19 | 2.51 | 1.23 | 0.488 | 0.122 | 0.024 | 0.015 |
| 34 | 7,656 | 2,120 | 119.2 | 23.8 | 19.9 | 13.2 | 8.51 | 4.11 | 2.02 | 0.800 | 0.199 | 0.040 | 0.024 |
| 50 | 20,496 | 4,200 | 161.0 | 32.2 | 26.8 | 17.9 | 11.50 | 5.55 | 2.73 | 1.081 | 0.269 | 0.054 | 0.032 |
| 63 | 38,618 | 8,322 | 153.1 | 30.6 | 25.5 | 17.0 | 10.94 | 5.28 | 2.60 | 1.028 | 0.256 | 0.051 | 0.031 |
| 95 | 108,375 | 32,960 | 108.5 | 21.7 | 18.1 | 12.1 | 7.75 | 3.74 | 1.84 | 0.728 | 0.181 | 0.036 | 0.022 |
| 120 | 243,557 | 33,012 | 206.9 | 41.4 | 34.5 | 23.0 | 14.78 | 7.14 | 3.51 | 1.389 | 0.346 | 0.069 | 0,041 |
| | | batch sizes | 1 | 5 | 6 | 9 | 14 | 29 | 59 | 149 | 598 | 3,000 | 5,000 |